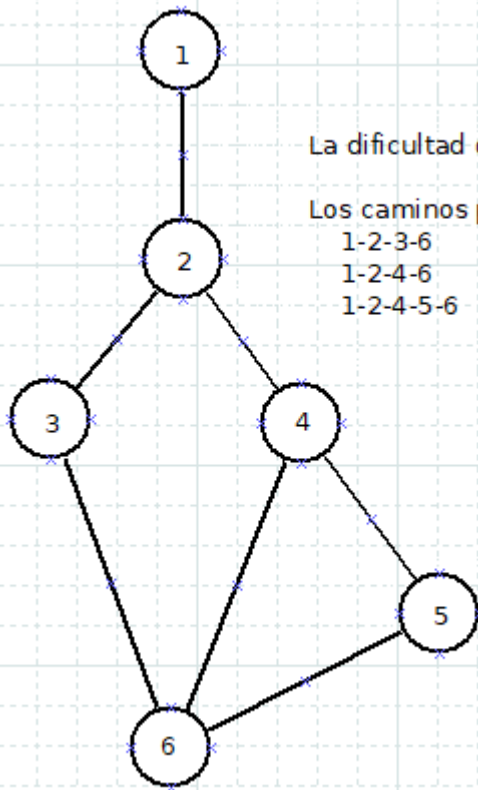


Ejercicio 1. Consultar Positivo



La dificultad de McCabe es de: $7 \text{ aristas} - 6 \text{ nodos} + 2 = 3$

Los caminos posibles son:

1-2-3-6

1-2-4-6

1-2-4-5-6

Ejercicio 2. Factura eléctrica

Nombre	Número	Resultado
CP1	-1	Error
CP2	0	9
CP3	300	9
CP4	301	8
CP5	600	8
CP6	601	6
CP7	1000	6
CP8	1001	5
CP9	2000	5
CP10	2001	Error

```
import static org.junit.Assert.assertEquals;
import org.junit.Test;

public class FacturaTest {

    @Test (expected = NumeroInvalidoException.class)
    public void CP1(){
        Factura fact = new Factura();
        fact.setConsumokWh(-1);
        fact.calcularPrezokWh();
    }

    @Test
    public void CP2() {
        Factura fact = new Factura();
        fact.setConsumokWh(consumokWh: 0);
        assertEquals(9, fact.calcularPrezokWh());
    }

    @Test
    public void CP3() {
        Factura fact = new Factura();
        fact.setConsumokWh(consumokWh: 300);
        assertEquals(9, fact.calcularPrezokWh());
    }

    @Test
    public void CP4() {
        Factura fact = new Factura();
        fact.setConsumokWh(consumokWh: 301);
        assertEquals(8, fact.calcularPrezokWh());
    }

    @Test
    public void CP5() {
        Factura fact = new Factura();
        fact.setConsumokWh(consumokWh: 600);
        assertEquals(8, fact.calcularPrezokWh());
    }

    @Test
    public void CP6() {
        Factura fact = new Factura();
        fact.setConsumokWh(consumokWh: 601);
        assertEquals(6, fact.calcularPrezokWh());
    }
}
```

```

@Test
public void CP7() {
    Factura fact = new Factura();
    fact.setConsumokWh(consumokWh: 1000);
    assertEquals(6, fact.calcularPrezokWh());
}

@Test
public void CP8() {
    Factura fact = new Factura();
    fact.setConsumokWh(consumokWh: 1001);
    assertEquals(5, fact.calcularPrezokWh());
}

@Test
public void CP9() {
    Factura fact = new Factura();
    fact.setConsumokWh(consumokWh: 2000);
    assertEquals(5, fact.calcularPrezokWh());
}

@Test (expected = NumeroInvalidoException.class)
public void CP10(){    java.lang.AssertionError: Expected exception: NumeroInvalidoException
    Factura fact = new Factura();
    fact.setConsumokWh(consumokWh: 2001);
    fact.calcularPrezokWh();
}
}

```

Hay errores en el código, las modificaciones requeridas son estas:

```

public int calcularPrezokWh(){
    int prezokWh=0;
    if(consumokWh < 0){
        throw new NumeroInvalidoException();
    }else if (consumokWh<=300){
        prezokWh=9;
    }else if(consumokWh<=600){
        prezokWh=8;
    }else if(consumokWh<=1000){
        prezokWh=6;
    }else if (consumokWh<=2000){
        prezokWh=5;
    }else{
        throw new NumeroInvalidoException();
    }
    return prezokWh;
}

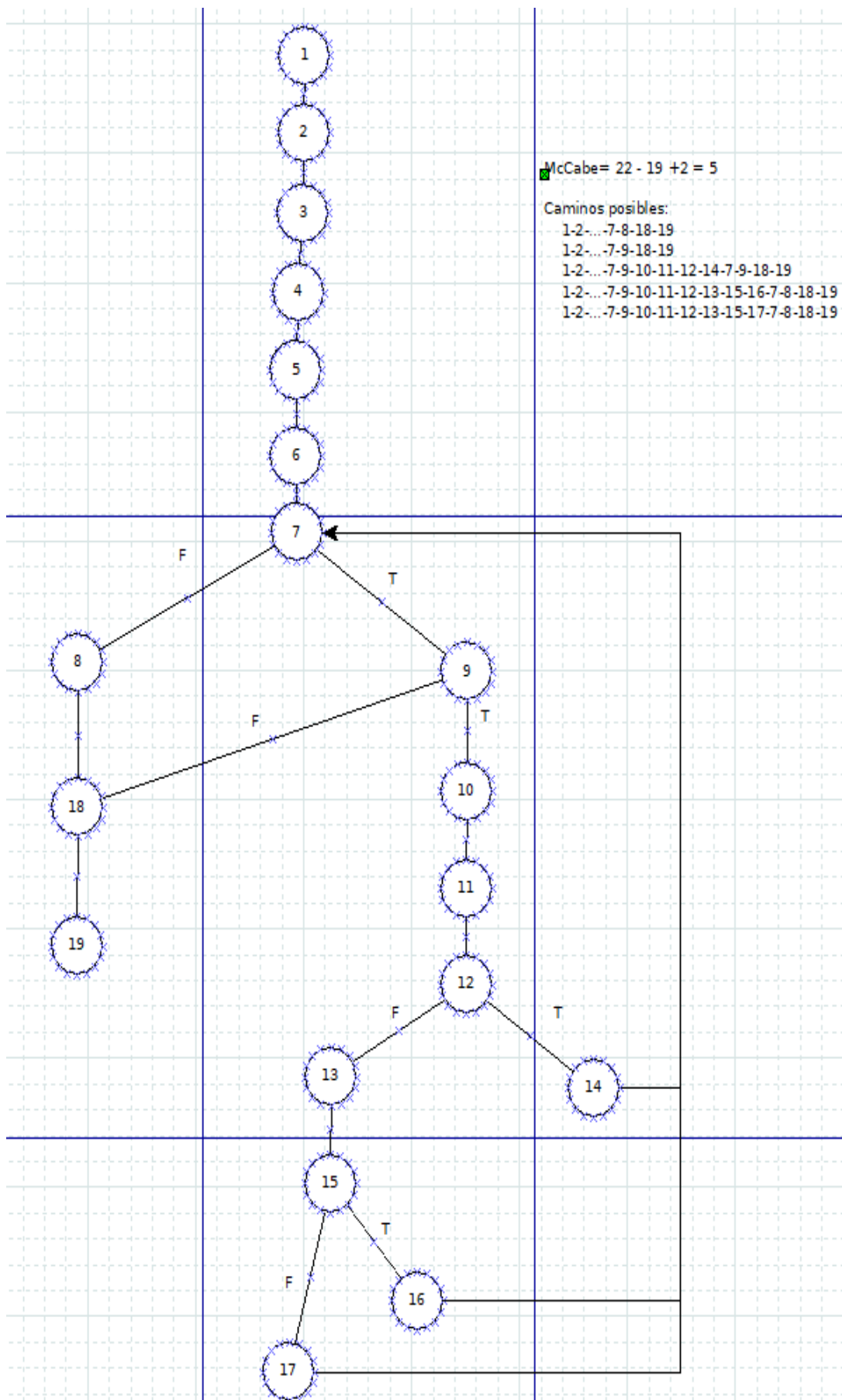
```

```

✓ ContornosDesenvolvimento
✓ <Default Package> 5.0ms
  ✓ FacturaTest 5.0ms
    ✓ CP10 4.0ms
    ✓ CP20 0.0ms
    ✓ CP30 0.0ms
    ✓ CP40 0.0ms
    ✓ CP50 0.0ms
    ✓ CP60 0.0ms
    ✓ CP70 0.0ms
    ✓ CP80 1.0ms
    ✓ CP90 0.0ms
    ✓ CP100 0.0ms

```

Ejercicio 3. Arrays



Ejercicio 4. Acrónimos

